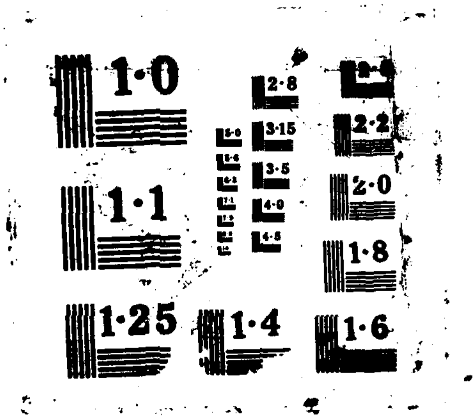
AD-A185 378 WIS (WWMCCS (WORLD WIDE MILITARY COMMAND AND CONTROL SYSTEM) INFORMATION S. (U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA R J KNAPPER APR 87 IDA-M-316 1/1

UNCLASSIFIED IDA/HQ-87-32154 MDA903-84-C-0031 F/G 12/5 NL


1·0
2·8
3·15
3·5
4·0
1·1
2·2
2·0
1·8
1·25
1·4
1·6

UNCLASSIFIED
AD-A185 378
ADE500 888
Copy 13 of 28 copies

(2)

IDA MEMORANDUM REPORT M-316

FINAL REPORT: WIS Ada* TRANSLATOR TESTING

Robert J. Knapper

April 1987

Prepared for
Office of the Under Secretary of Defense for Research and Engineering

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

*Ada is a registered trademark of the U.S. Government, Ada Joint Program Office.

UNCLASSIFIED
IDA Log No. HQ 87-32154
87 8 17 070

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, or (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to insure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Papers

Papers normally address relatively restricted technical or policy issues. They communicate the results of special analyses, interim reports or phases of a task, ad hoc or quick reaction work. Papers are reviewed to insure that they meet standards similar to those expected of referred papers in professional journals.

Memorandum Reports

IDA Memorandum Reports are used for the convenience of the sponsors or the analysts to record substantive work done in quick reaction studies and major interactive technical support activities; to make available preliminary and tentative results of analyses or of working group and panel activities; to forward information that is essentially unanalyzed and unevaluated; or to make a record of conferences, meetings, or briefings, or of data developed in the course of an investigation. Review of Memorandum Reports is suited to their content and intended use.

The results of IDA work are also conveyed by briefings and informal memoranda to sponsors and others designated by the sponsors, when appropriate.

The work reported in this document was conducted under contract MDA 903 84 C 0031 for the Department of Defense. The publication of this IDA Memorandum Report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that agency.

This Memorandum Report is published in order to make available the material it contains for the use and convenience of interested parties. The material has not necessarily been completely evaluated and analyzed, nor subjected to IDA review.

REPORT DOCUMENTATION PAGE

AD-A185 378

1a REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Public release/unlimited distribution.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) M-316	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses 6b OFFICE SYMBOL	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and Zip Code) 1801 N. Beauregard St. Alexandria, VA 22311	7b ADDRESS (City, State, and Zip Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION WIS JPMO/ADT (OUSDRE/C3I) 8b OFFICE SYMBOL (if applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA 903 84 C 0031
8c ADDRESS (City, State, and Zip Code) WIS JPMO/ADT Washington, D.C. 20330-6600	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: PROJECT NO.: TASK NO.: T-W5-260 WORK UNIT ACCESSION NO.:

11 TITLE (Include Security Classification)
Final Report: WIS Ada Translator Testing (U)

12 PERSONAL AUTHOR(S)
Robert J. Knapper

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM ______ TO ______	1987 April	40

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Ada Programming Language; Worldwide Military Command and Control System (WWMCCS); WWMCCS Information System (WIS); Software; Tools; FORTRAN 66; COBOL 68; Translation; Prototypes; Acceptance Testing; code.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The purpose of this IDA Memorandum Report is to forward data developed in the course of acceptance testing on the FORTRAN and COBOL to Ada translators constructed for the WIS JPMO. The major objective in the WWMCCS translator project is to develop a set of production quality translation tools. These tools are to provide 100% automated translation into MIL-STD-1815A Ada of the legal, standard features of one dialect each of FORTRAN 66 and COBOL 68 common to WWMCCS ADP software. Ada is the language to be used in constructing WIS to enhance software component portability and to increase machine independence.

By reusing WWMCCS ADP software through translation to Ada, the costs to redesign and re-engineer portions of the joint mission software will be deferred.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ■ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) / 22c OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

UNCLASSIFIED

IDA MEMORANDUM REPORT M-316

FINAL REPORT: WIS Ada* TRANSLATOR TESTING

Robert J. Knapper

April 1987

INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
Task T-W5-260

UNCLASSIFIED

TABLE OF CONTENTS

PREFACE

The purpose of IDA Memorandum Report M-316, "Final Report - WIS Ada Translator Testing", is to forward data developed in the course of performing acceptance testing on the FORTRAN and COBOL to Ada translators constructed for the World Wide Military Command and Control Information System Joint Program Management Office (WIS JPMO).

The importance of this document is based on fulfilling the objective of Task Order T-W5-260, WIS Ada Translator, which is to test and document the acceptance of the translators. M-316 will be used to provide evidence of the effort in confirming the functionality of the translators delivered. As a Memorandum Report, M-316 is directed towards the potential users of the translators.

The preliminary document was reviewed in September 1986 and the final document was revived in January 1987 by Dr. John F. Kramer and Dr. John Salasin. An external review was also performed by Lt. Col. Terry Courtwright and Capt. Jacques Choiniere, WIS JPMO.

Final Report - WIS Ada* Translator Testing
Task T-W5-260

Robert J. Knapper
Institute for Defense Analyses
Computer and Software Engineering Division
1801 N. Beauregard Street
Alexandria, VA 22311

1.0 ABSTRACT

The Worldwide Military Command and Control System (WWMCCS) has as a component the data collection and processing facilities of our defense system. This component, WWMCCS ADP, was installed by Honeywell Information Systems starting in 1971. The application software for the H6000 machines was written in FORTRAN 66 and COBOL 68. In the past 15 years, that software has been patched and updated, with new modules in different FORTRAN and COBOL dialects added.

A modernization of WWMCCS ADP to the WWMCCS Information System (WIS) was started in 1982. The software for WIS, unlike that for WWMCCS ADP, is intended to be independent of the hardware on which it is to run. The use of MIL-STD-1815A Ada will support the construction of machine independent software.

To assist in bringing WIS on-line faster and to help defer the costs in developing new WIS software in Ada, automated tools to translate existing WWMCCS ADP software from the original FORTRAN and COBOL to Ada have been constructed.

2.0 INTRODUCTION

The major objective in the WWMCCS translator project is to develop a set of production quality translation tools. These tools are to provide 100 percent automated translation into MIL-STD-1815A Ada of the legal, standard features of one dialect each of FORTRAN 66 and COBOL 68 common to WWMCCS ADP software. Ada is the language to be used in constructing WIS to enhance software component portability and to increase machine independence. By reusing WWMCCS ADP software through translation to Ada, the costs to redesign and reengineer portions of the joint mission software will be defered. This will allow an integrated, functioning WIS that is machine independent to be brought on-line and into use faster. Eventually, all of the WWMCCS ADP software will be redesigned and recoded into Ada, but the use of the interim capability through source translation in order to defer costs is desirable. The need for defering costs is based on the fact that WWMCCS ADP joint software may be as large as 20 millions lines of code. All of this code cannot be redesigned immediately.

3.0 BACKGROUND

WWMCCS consists of five major elements: communication systems, early warning systems, command facilities, executive aids, and data collection and processing. The data collection and processing element, WWMCCS ADP, provides the information processing support needed by military planners and operators in planning military operations, monitoring plan execution, and replanning in response to changing military situations.

WWMCCS ADP hardware and system software was installed by Honeywell Information Systems starting in 1971. In the past 15 years, the initial installation of H6000 hardware and GCOS III operating systems for WWMCCS ADP has grown to include over 100 CPUs in 34 different locations. Large volumes of both joint mission software and command unique software are executed at each location.

The evolution of WWMCCS ADP applications around the Honeywell H6000 and GCOS III base has led to a severe dependence on out-dated equipment resulting in software that is difficult to upgrade and maintain. Recognizing this dependence, a plan to modernize WWMCCS ADP into WIS was approved in 1982. The approach to be taken in the modernization includes the key concept that applications software can be developed that is independent of a particular hardware and or environment. Therefore, development, testing, and implementation of joint mission software modules can be done without the need to depend upon future joint mission hardware acquisitions. The use of Ada, as a closely controlled military, national, and international standard, for the design and software engineering of WIS will support the goal of machine independence for WIS software.

Some of the capabilities in WWMCCS ADP software will continue to be required in WIS. The redesign and reengineering of these capabilities is an eventuality, however an interim reengineering through the use of source to source translation will not only assist in bringing WIS on-line faster, but will also defer the complete reengineering costs and will expedite the use of Ada in WIS.

The WIS modernization effort is under the management control of the WIS Joint Program Management Office (JPMO). In support of the WIS JPMO, the Institute for Defense Analyses (IDA) has been tasked to provide a set of translation tools. IDA, through a subcontract to Advanced Technology Systems (ATS), began work on the tool set in January 1985 and delivered the final products in November 1986.

4.0 TRANSLATION TOOLS

The FORTRAN and COBOL translators are actually a collection of five translation tools:

a. FORTRAN Source Analyzer - analyzes WWMCCS ADP FORTRAN source and creates an Attributed Abstract Syntax Tree (AAST).

b. FORTRAN Tree Transformer - transforms the FORTRAN AAST into Descriptive Intermediate Attributed Notation for Ada (DIANA).

c. COBOL Source Analyzer - analyzes WWMCCS ADP COBOL source and creates an AAST.

d. COBOL Tree Transformer - transforms the COBOL AAST into DIANA.

e. DIANA Converter - creates Ada textual representations from the DIANA form. Also provides minimal aesthetic formatting.

These components are supported by FORTRAN and COBOL pattern tables and by run-time support packages to be used by the translated code during compilation and execution.

4.1 Design Approach

The approach taken for the translator construction is based on current compiler construction technology; a front-end source analyzer producing an abstract syntax tree, a middle-end to walk the tree and generate an intemediate langauge, and finally a back-end code generator. The tree is produced by the analyzers are extended by using mappings of the source language as patterns. The resulting tree is a DIANA tree representing the equivalent Ada program.

5.0 ACCEPTANCE TESTING

The testing of the translation tools was a three phase process. First, the individual prototypes were tested for functionality and completeness of the features specified for that prototype. Second, the overall conformity of the final prototypes was checked with compiler test suites, and lastly, the final prototypes were required to successfully translate live WWMCCS ADP software using generated or declassified, live data.

5.1 Prototype Functionality

The functionality of each prototype was checked by exercising each of the features and combination of features to be supported by that prototype. ATS provided an on-line set of tests that were used to check prototype features. IDA used this set plus modified and added to the set to check for prototype functionality. Since succeeding prototypes are built upon previous ones, a particular prototype must pass not only the new tests but any previous test also. There is no contingent passing of the tests. The prototype must function fully as specified or it is not accepted.

All of the 11 Prototypes (5 FORTRAN and 6 COBOL, see Appendix A) passed these tests at the level specified. The tests used to validate the last two delivered prototypes, P10-FORTRAN and P11-COBOL are included with the translator source and documentation set. They incorporate all of the previous features required by the translator.

5.2 Conformity Testing

The General Services Administration Software Testing Center maintains test suites for FORTRAN and COBOL compiler testing. These tests allow conformity to the FORTRAN 66 and COBOL 68 standards to be checked. The suites contain syntactically and semantically correct programs that are self-supporting. They generate their own data and produce a report file when compiled and executed. Properly executing programs so indicate in the report. These programs were translated into Ada with the tools then compiled and executed. The results generated by the translated versions are compared with the original versions electronically except where noted in the testing record (Appendixes B and C).

5.2.1 COBOL Test Suite

The COBOL test suite consists of 319 test programs in 12 modules:

-- Communication
-- Debug
-- Inter-Program Communication
-- Indexed I/O
-- Library
-- Nucleus
-- Relative I/O
-- Report Writer
-- Segmentation
-- Sequential I/O
-- Sort-Merge
-- Table Handling

Of the 319 programs, 193 were applicable to the particular dialect of COBOL used as the source for the translation. Most of the remaining 126 tests come from the Debug module, the Report Writer module, and from I/O features that are not supported by the translator. Two of the 193 tests were modified to remove an optional feature of COBOL that the translator does not support. The 193 programs contained nearly 30,000 lines of COBOL source and thoroughly exercised the capability of the translator to recognize and correctly translate COBOL syntax for the dialect used. The translator passed these tests (see Appendix B).

5.2.2 FORTRAN Test Suite

The FORTRAN test suite contains 193 test programs in 3 categories

-- Subset 1
-- Subset 2
-- Full Language

Of the 193 programs, 105 were applicable to the particular dialect of FORTRAN used as the source for the translation. Most of the 88 tests not applicable come from the Full Language category. The sophisticated and intricate interfaces and interactions between features and program semantics are not handled by this particular dialect. Of the 105 applicable programs 28 required inspection to determine the results. This was because the FORTRAN intrinsic mathematical functions (SIN, COS, EXP, etc.) were correctly translated as Ada function calls, but the translated Ada code needed to be edited to include access to an Ada mathematical library package. In addition, one test was modified to remove an optional part of FORTRAN that the translator does not support. The 105 programs contained over 15,000 lines of FORTRAN source and thoroughly exercised the capability of the translator to recognize and correctly translate FORTRAN syntax. The translator passed these tests (see Appendix C).

5.2.3 Coverage by Test Suites

The non-applicability of 126 of the 319 COBOL tests and 88 of the 193 FORTRAN tests should not be viewed as a weakening of this testing effort. The GSA suites are designed to be tailored to particular applications and contain repetitive code as well as code for features that the translator was not contracted to process.

5.3 Final Testing

WWMCCS ADP software was used for the final acceptance testing of the tools. The software, using random data generated with automated tools or live, declassified data, was executed on the H6000 system. The results of this execution were captured in a file. The system was then translated into Ada, compiled and executed with the same data set and the results were compared with the original results. The translated version was required to be functionally identical in all respects.

Code from JOPS III (COBOL) and RapidSim (FORTRAN), plus additional COBOL and FORTRAN modules of WWMCCS code were run through the translator. The tool passed these tests (see Appendix D).

In total (GSA suites + WWMCCS code), approximately 110,000 lines of COBOL and FORTRAN source code was used in testing the translator.

6.0 Conclusion

The modernization of WWMCCS ADP software into WIS through redesign and reengineering is underway and will eventually be completed. In the interim, some functional capabilities of WWMCCS ADP can be transitioned to WIS more rapidly through the use of source to source translation. In effect, the old code is reused as portable, machine independent Ada modules thus reducing the time to bring an integrated, functioning WIS on-line while defering the costs for the complete redesign and implementation. In addition, Ada, which is a DoD mandated language for mission critical systems, is not only expeditiously utilized, but is promoted for further use in software engineering throughout DoD and industry.

The WIS Ada Translation tool, produced for the WIS JPMO will successfully translate COBOL and FORTRAN source code if used properly. The information in the user's guide provided with the tool must be followed to the letter, otherwise the tool will "hang up" in execution, or will not produce a translation at all. While following the rules may involve some changes to the original source, merging of certain files, and modification of I/O interfaces (all of which can be time consuming and potentially frustrating), the alternative of completely re-writing the source in Ada would take considerably more time.

The execution of the translator is slow. On a VAX 8600 with a small load and plenty of available resources, the tool will take up to one hour of CPU time (and over one hour real time) to translate 1500 lines of complex code.

Finally, the translator does not provide (nor was it intended to provide) a fully automatic, turn-key, "black box" for translating code. It does however, provide 100% automatic translation of the correct and legal features of COBOL and FORTRAN as specified and contracted for. The best word to remember when using the translator is patience.

APPENDIX A

Translator Prototypes Delivery Schedule*

Deliverable		Date	Language
Prototype 1	Scheduled	April 12, 1985	FORTRAN
	Actual	April 12, 1985	
Prototype 2	Scheduled	July 12, 1985	FORTRAN
	Actual	July 12, 1985	
Prototype 3	Scheduled	Sept 13, 1985	COBOL
	Actual	Sept 13, 1985	
Prototype 4	Scheduled	Oct 14, 1985	FORTRAN
	Actual	Oct 17, 1985	
Prototype 5	Scheduled	Dec 13, 1985	COBOL
	Actual	Jan 20, 1986	
Prototype 6	Scheduled	Jan 14, 1986	FORTRAN
	Actual	Jan 20, 1986	
Prototype 7	Scheduled	March 14, 1986	COBOL
	Actual	April 21, 1986	
Prototype 8	Scheduled	April 14, 1986	FORTRAN
	Actual	April 21, 1986	
Prototype 9	Scheduled	June 13, 1986	COBOL
	Actual	July 30, 1986	
Prototype 10	Scheduled	July 14, 1986	FORTRAN
	Actual	Nov 17, 1986	
Prototype 11	Scheduled	Sept 14, 1986	COBOL
	Actual	Oct 15, 1986	

*There were originally 10 prototypes planned. The 11th prototype was agreed upon to provide a nearly complete COBOL capability as close to the original Prototype 9 completion date as was possible.

A move was made from a Data General MV10000 system to a VAX 8600 system in January 1986. This resulted in a one month overall delay in the schedule.

APPENDIX B

COBOL Suite Testing

P = PASSED, F=FAILED, PM = PASSED WITH MODIFICATION,
NA = NOT APPLICABLE

TEST	RESULT	COMMENTS
CM101	P	
CM102	P	
CM103	P	
CM104	P	
CM105	P	
CM201	P	
CM202	NA	Features not supported
CM431	P	COMP-FLAG = 1
DB101-DB105, DB201-DB205, DB421-DB422	NA	Features not supported
IC101	P	
IC102	P	
IC103	P	
IC104	P	
IC105	P	
IC106	P	
IC107	P	
IC108	P	
IC109	P	
IC110	P	
IC111	P	

IC112	P	
IC113	P	
IC114	P	
IC115	P	
IC116	P	
IC117	P	
IC118	P	
IC151	P	
IC152	P	
IC201	P	
IC202	P	
IC203	P	
IC204	P	
IC205	P	
IC206	P	
IC207	P	
IC208	P	
IC209	P	
IC210	P	
IC211	P	
IC212	P	
IC213	P	
IC214	P	
IC215	P	
IC216	P	
IC217	P	
IC218	P	

IC219	P	
IC220	P	
IC421	P	
IC422	P	
IC431	P	
IC432	P	
IX101-IX107, IX201-IX215, IX441-IX442	NA	Features not supported
LB101-LB107, LB201-LB207, LB421+LB442	NA	Features not supported
NC101	P	
NC102	P	
NC103	P	
NC104	P	
NC105	P	
NC106	P	
NC107	PM	CURRENCY SIGN IS removed
NC108	PM	CURRENCY IS removed
NC109	P	
NC110	P	
NC111	P	
NC112	P	
NC113	P	
NC114	P	
NC115	P	
NC116	P	
NC117	P	

NC118	P	
NC119	P	
NC120	P	
NC121	P	
NC122	P	
NC123	P	
NC124	P	
NC125	P	
NC151	P	
NC152	P	
NC153	P	
NC154	P	
NC155	P	
NC156	P	
NC157	P	
NC158	P	
NC159	P	
NC160	P	
NC161	P	
NC162	P	
NC163	P	
NC164	P	
NC165	P	
NC201	P	
NC202	P	
NC203	P	
NC204	NA	Features not supported

NC205	P	
NC206	P	
NC207	P	
NC208	P	
NC209	P	
NC210	P	
NC211	P	
NC212	P	
NC213	P	
NC214	P	
NC215-NC221	NA	Features not supported
NC222	P	
NC431	P	
RL101	P	
RL102	P	
RL103	P	
RL104-RL107, RL151-RL153, RL201-RL204, RL421+RL431	NA	Features not supported
RW101-RW104	NA	Features not supported
SG101	P	
SG102	P	
SG103	P	
SG104	P	
SG105	P	
SG106	P	
SG201-SG202	NA	Features not supported

SG204	NA	Features not supported
SG421	NA	Features not supported
SG431	NA	Features not supported
SQ101	P	
SQ102	P	
SQ103	P	
SQ104	P	
SQ105	P	
SQ106	P	
SQ107	P	
SQ108	P	
SQ109-SQ111	NA	Features not supported
SQ112	P	
SQ113	P	
SQ114-SQ121, SQ151-SQ153, SQ201-SQ202	NA	Features not supported
SQ203	P	
SQ204-SQ216	NA	Features not supported
SQ217	P	
SQ218	NA	Features not supported
SQ219	P	
SQ431	P	
SQ432	P	
ST101	P	
ST102	P	
ST103	P	
ST104	P	

ST105	P	
ST106	P	
ST107	P	
ST108	P	
ST109	P	
ST110	P	
ST111	P	
ST112	NA	Features not supported
ST113	P	
ST114	P	
ST115	P	
ST116	P	
ST117	P	
ST118	P	
ST201	P	
ST202	NA	Features not supported
ST203	P	
ST204	P	
ST205-ST206	NA	Features not supported
ST207	P	
ST208	NA	Features not supported
ST209	P	
ST210	P	
ST211	P	
ST212	P	
ST213	P	
ST214	P	

ST215-ST216, ST431-ST432, ST433-ST434, ST441-ST443	NA	Features not supported
TH101	P	
TH102	P	
TH103	P	
TH104	P	
TH105	P	
TH106	P	
TH107	P	
TH108	P	
TH109	P	
TH110	P	
TH111	P	
TH112	P	
TH151	P	
TH152	P	
TH201	P	
TH202	P	
TH203	P	
TH204	P	
TH205	P	
TH206	P	
TH207	P	
TH208	P	
TH209	P	
TH210	P	

TH211	P	
TH212	P	
TH213	P	
TH214	P	
TH215	P	
TH216	P	
TH217	P	
TH218	P	
TH219	P	
TH220	P	
TH221	NA	Features not supported
TH222	P	
TH431	P	

Total Tests	319
Passed	191
Passed with Modification	2
Failed	0
Not Applicable	127

APPENDIX C

FORTRAN Suite Testing

P - PASSED, F - FAILED, PI - PASSED BY INSPECTION,
PM - PASSED WITH MODIFICATION, NA - NOT APPLICABLE

TEST	RESULT	COMMENTS
FM001	P	
FM002	P	
FM003	P	
FM004	P	
FM005	P	
FM006	P	
FM007	P	
FM008	P	
FM009	P	
FM010	P	
FM011	P	
FM012	P	
FM013	P	
FM014	P	
FM016	P	
FM017	P	
FM018	P	
FM019	P	
FM020	P	
FM021	P	

FM022	P	
FM023	P	
FM024	P	
FM025	P	
FM026	P	
FM028	P	
FM030	P	
FM031	P	
FM032	P	
FM033	P	
FM034	P	
FM035	P	
FM036	P	
FM037	P	
FM038	P	
FM039	P	
FM040	P	
FM041	P	
FM042	P	
FM043	P	
FM044	P	
FM045	P	
FM050	P	
FM056	P	
FM060	P	
FM061	P	
FM062	P	

FM080	P	
FM097	PI	Need Ada math package for intrinsic fcns
FM098	PI	Need Ada math package for intrinsic fcns
FM099	PI	Need Ada math package for intrinsic fcns
FM100	P	
FM101	P	
FM102	P	
FM103	P	
FM104	P	
FM105	P	
FM106	P	
FM107	P	
FM108	P	
FM109	P	
FM110	P	
FM111	P	
FM200-FM201	NA	Features not supported
FM202	P	
FM203	NA	Features not supported
FM204	PI	Need Ada package for intrinsic chr function
FM205	NA	Features not supported
FM250	NA	Features not supported
FM252	PM	Optional comma test removed
FM253-FM261	NA	Features not supported
FM300	P	
FM301	P	
FM302	P	

FM306	NA	Features not supported
FM307	PI	Need Ada math package for intrinsic fcns
FM308	NA	Features not supported
FM311	NA	Features not supported
FM317	P	
FM328	P	
FM351	P	
FM352	P	
FM353-FM357, FM359-FM364, FM368-FM379	PI	Need Ada math package for intrinsic fcns
FM401	P	
FM402	P	
FM403	P	
FM404	P	
FM405	P	
FM406	P	
FM407	P	
FM411	P	
FM413, FM500, FM503, FM506, FM509, FM514, FM517, FM520, FM700, FM701, FM710, FM711, FM715, FM718, FM719, FM722, FM800-FM834, FM900, FM901, FM903, FM905, FM906-FM910, FM912, FM914, FM915-FM923		Features not supported

Total Tests	193
Passed	76
Passed by Inspection	28
Passed with Modification	1
Failed	0
Not Applicable	88

Appendix D

WWMCCS Code Testing

JOPS III Modules Tested (COBOL)

TEST	RESULT	COMMENTS
AIRDL	P	
AIRSUM	P	
AIRTP	P	
CNTRL	P	
FM	P	
FM01	P	
FM01A	P	
FM01B	P	
FM01C	P	
FM01D	P	
FM02	P	
FM03	P	
FM04	P	
FM05	P	
FM06	P	
FORCE	P	
F10	P	
F10A	P	
F10B	P	
F10C	P	
F10D	P	

F10E	P	
F11	P	
F11D	P	
F11E	P	
F11F	P	
F11G	P	
F11T	P	
F11W	P	
F11X	P	
F12A	P	
F12B	P	
F12B1	P	
F12C	P	
F12D	P	
F12E	P	
F12E1	P	
F12G	P	
F12H	P	
F12K	P	
F12KK	P	
F12L	P	
F12M	P	
F12P	P	
F12R	P	
F12S	P	
F12T	P	
F12W	P	

F12X	P	
F12Y	P	
F14	P	
F14A	P	
F14G	P	
F14J	P	
F14M	P	
F14R	P	
F14S	P	
F15	P	
F15A	P	
F15B	P	
F20	P	
F21	P	
F21A	P	
F21B	P	
F22	P	
F23	P	
F23A	P	
F23B	P	
F24	P	
F24A	P	
F24B	P	
F25	P	
F26	P	
F26A	P	
F30	P	

F35	P	
F38	P	
F40	P	
F40A	P	
F40D	P	
F40E	P	
F50	P	
F53	P	
F60	P	
F60A	P	
F60B	P	
F60C	P	
F60D	P	
F60P	P	
F61	P	
F61A	P	
F61C	P	
F61D	P	
F61F	P	
F61H	P	
F70	P	
F71	P	
F71A	P	
F72	P	
F80	P	
GEORPT	P	

Using the COBOL translator on WWMCCS code required preparatory modifications on every file. All references to data files other than flat, sequential ASCII files were removed. Any GMAP code, calls to equipment through non-standard COBOL interfaces, etc. were also removed. The translated neutered code performed identically to the neutered untranslated code. The use of the COBOL translator for real WWMCCS code will be restrictive because of these modification needs. However, if follow-on extensions to the translator are funded and completed, database access, GMAP interaction, etc. will be less of a problem.

P - PASSED, F - FAILED

TEST	RESULT	COMMENTS
AIRDST.FOR	P	
ARANDM.FOR	P	
ARRIVE.FOR	P	
ASGCGO.FOR	P	
ATHOLD.FOR	P	
ATPOOL.FOR	P	
BEGIN.FOR	P	
CALCRT.FOR	P	
CARDEV.FOR	P	
CARTR.FOR	P	
CENTER.FOR	P	
CLKADJ.FOR	P	
CNVRT.FOR	P	
CNVYGO.FOR	P	
CONRPT.FOR	P	
CONTRL.FOR	P	
CONVRT.FOR	P	
DATIME.FOR	P	
DELAY.FOR	P	
DEPART.FOR	P	
DISTRB.FOR	P	
ERROR1.FOR	P	

ERRORS.FOR	P	
EVTGEN.FOR	P	
EVTRPT.FOR	P	
FILDEF.FOR	P	
FINDTS.FOR	P	
GAUSS.FOR	P	
HEAD.FOR	P	
HEAD1.FOR	P	
INPROC.FOR	P	
INTER.FOR	P	
INTMOG.FOR	P	
LEADER.FOR	P	
LISP.FOR	P	
LNKDST.FOR	P	
LSTPRT.FOR	P	
MODNOD.FOR	P	
MOMEN.FOR	P	
MUSTGO.FOR	P	
NAMES.FOR	P	
NETEVT.FOR	P	
NETIN.FOR	P	
NEWCHK.FOR	P	
NEXJOB.FOR	P	
NEXPOD.FOR	P	
NORPO.FOR	P	
NSHIFT.FOR	P	
NSREPT.FOR	P	

NTWEXT.FOR	P	
NTWORK.FOR	P	
NWPAGE.FOR	P	
OEVENT.FOR	P	
ORDSCH.FOR	P	
OUPROC.FOR	P	
PAGHED.FOR	P	
PUTOUT.FOR	P	
RANDM.FOR	P	
RATER.FOR	P	
READ1.FOR	P	
READIN.FOR	P	
REEDIN.FOR	P	
REROUT.FOR	P	
RESADJ.FOR	P	
RESAVL.FOR	P	
RESMOG.FOR	P	
RPPRNT.FOR	P	
RPT1.FOR	P	
RPT2.FOR	P	
RPT3.FOR	P	
RTRPT.FOR	P	
SEGACT.FOR	P	
SEGASG.FOR	P	
SEGLIM.FOR	P	
SELECT.FOR	P	
SETDIS.FOR	P	

SEVENT.FOR	P	
SHPRPRT.FOR	P	
SITAP1.FOR	P	
SITAP1.FOR	P	
SITAP2.FOR	P	
SITAP3.FOR	P	
SLDMOG.FOR	P	
SPILL.FOR	P	
STAGE.FOR	P	
SUPER.FOR	P	
TASKER.FOR	P	
TDISF.FOR	P	
UEVENT.FOR	P	
UTRACE3.FOR	P	
WRAPUP.FOR	P	
WRITEM.FOR	P	
WRTSHP.FOR	P	
ZCKMOG.FOR	P	

The FORTRAN translator requires that every file presented to it be a complete program, i.e. a subprogram or several subprograms in a single file must be accompanied by a main program in that same file. There are two ways to approach this problem. The first is to include the entire system all in one file. This is neither feasible nor wise. Too many resources, (CPU time most notably) would be needed. The second way is to give the subprograms "dummy" main programs. These programs are then translated into Ada. The Ada files are edited to make the dummy main programs into packages containing the subprograms. References can the be made to the compiled routines within the packages as if they were FORTRAN library subprograms. This is the recommended approach.

Distribution List for IDA Memorandum Report M-316

NAME AND ADDRESS	NUMBER OF COPIES
Sponsors	
Dr. Michael Mendiville WIS JPMO/ADT The Pentagon Washington, DC 20330-6600	5 copies
Other	
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2 copies
CSED Review Panel	
Dr. Dan Alpert, Director Center for Advanced Study University of Illinois 912 W. Illinois Street Urbana, Illinois 61801	1 copy
Dr. Barry W. Boehm TRW Defense Systems Group MS 2-2304 One Space Park Redondo Beach, CA 90278	1 copy
Dr. Ruth Davis The Pymatuning Group, Inc. 2000 N. 15th Street, Suite 707 Arlington, VA 22201	1 copy
Dr. Larry E. Druffel Software Engineering Institute Shadyside Place 480 South Aiken Av. Pittsburgh, PA 15231	1 copy
Dr. C.E. Hutchinson, Dean Thayer School of Engineering Dartmouth College Hanover, NH 03755	1 copy
Mr. A.J. Jordano Manager, Systems & Software Engineering Headquarters Federal Systems Division 6600 Rockledge Dr. Bethesda, MD 20817	1 copy

Mr. Robert K. Lehto Mainstay 302 Mill St. Occoquan, VA 22125	1 copy
Mr. Oliver Selfridge 45 Percy Road Lexington, MA 02173	1 copy

IDA

General W.Y. Smith, HQ	1 copy
Mr. Seymour Deitchman, HQ	1 copy
Ms. Karen H. Weber, HQ	1 copy
Dr. Jack Kramer, CSED	1 copy
Dr. Robert I. Winner, CSED	1 copy
Dr. John Salasin, CSED	1 copy
Mr. Robert J. Knapper, CSED	2 copies
Ms. Katydean Price, CSED	2 copies
IDA Control & Distribution Vault	3 copies

END
11-87
DTIC